HIPRank: Ranking Nodes by Influence Propagation Based on Authority and Hub

Wen Zhang, Song Wang, Guangle Han, Ye Yang, and Qing Wang

Abstract Traditional centrality measures such as degree, betweenness, closeness and eigenvector ignore the intrinsic impacts of a node on other nodes. This paper proposes a new algorithm, called HIPRank, to rank nodes based on their influences in the network. HIPRank includes two sub-procedures: one is to predefine the importance of an arbitrary small number of nodes with users' preferences, and the other one is to propagate the influences of nodes with respect to authority and hub to other nodes based on HIP propagation model. Experiments on DBLP citation network (over 1.5 million nodes and 2.1 million edges) demonstrate that on the one hand, HIPRank can prioritize the nodes having close relation to the user-preferred nodes with higher ranking than other nodes, and on the other hand, HIPRank can retrieve the authoritative nodes (with authority) and directive nodes (with hub) from the network according to users' preferences.

Keywords Influence propagation • Network modeling • Nodes ranking

1 Introduction

Recently, network-based search arises in both research and application areas. Those traditional algorithms, such as Google's PageRank [8] and Kleinberg's HITS [5], assumed a global view on the structure of the network to treat all the users' preferences [4] in ranking nodes equally. However, in most cases, this

W. Zhang (✉)
School of Management and Economics, Beijing University of Chemical Technology, Beijing, China
e-mail: zhangwen@mail.buct.edu.cn

S. Wang
Chinese Academy of Sciences, University of Chinese Academy of Sciences, China
e-mail: wangsong@nfs.iscas.ac.cn

G. Han • Y. Yang • Q. Wang
Chinese Academy of Sciences, Beijing, China
e-mail: guangle@nfs.iscas.ac.cn; yangye@nfs.iscas.ac.cn; wq@nfs.iscas.ac.cn

Q. Bai et al. (eds.), *Smart Modeling and Simulation for Complex Systems*, Studies in Computational Intelligence 564, DOI 10.1007/978-4-431-55209-3_1

assumption may be inappropriate because of the actual differences of users' personal preferences in nodes ranking. Thus, PPR (Personalized PageRank) [1, 2, 4] has been proposed to solve the problem of personal preferences. Although most of the research endeavor has been invested in speeding up the PPR to make its computation practical [1, 2, 4], similar to Pei Li et al [6], we attempted a different treatment to considering users' personal preferences by propagating the user-predefined importance of nodes in the network. Inspired by the Hyperlink-Induced Topic Search (HITS) [5], which measures the importance of web pages in the web network using *authority* and *hub*, we believe that connected nodes in the network can influence each other and the influence of nodes should be propagated bidirectionally (forward and backward) rather than unidirectional propagation in Pei Li et al [6].

The main contributions of this paper can be summarized in two aspects. First, we argue that the user preference or prior knowledge of nodes should be taken into consideration when ranking nodes in the network. Second, we propose a new influence propagation model called HIP to describe the bidirectional propagation of predefined importance of *authority* and *hub* over nodes accompanied with random walk paths. Based on these two aspects, we propose a new algorithm called HIPRank to rank individuals in the network. The computation complexity of our proposed algorithm HIPRank is much less than those proposed to solve the PPR problem.

The remainder of this paper is organized as follows. Section 2 describes the motivation. Section 3 presents the related work. Section 4 proposes the HIP model and HIPRank algorithm for ranking nodes in the network. Section 5 conducts experiments. Section 5.3 concludes the paper. Table 1 lists the notations used in this paper.

Table 1 Notations used in this paper

$G(V, E)$	G is a directed graph; V is the set of vertices of G; E is the set of edges of G
Z_a	The vector of predefined *authority* importance of all nodes
Z_h	The vector of predefined *hub* importance of all nodes
$Z_{a,i}(b)$	The influence of *authority* received by node b on the i-th propagation step
$Z_{h,i}(b)$	The influence of *hub* received by node b on the i-th propagation step
$Z_a(p, q)$	The influence of *authority* propagated from node p to q through all the paths from node p to q
$Z_h(p, q)$	The influence of *hub* propagated from node p to q through all the paths from node p to q
K	The maximum propagation step in HIP model
N	The nodes size of a directed graph
M	The edges size of a directed graph
P	The number of edges traversed in the propagation process
R_a	The *authority* ranking vector of all nodes
R_h	The *hub* ranking vector of all nodes
$R_{a,i}$	The *authority* ranking vector of all nodes on the i-th iteration/step
$R_{h,i}$	The *hub* ranking vector of all nodes on the i-th iteration/step
$O(i)$	The out-going neighbors of node i

2 The Motivation

Let W be the adjacency matrix of a directed graph G, and T is the transpose of W. For the element at p-th row and q-th column of W, in HITS, $W_{p,q} = 0$ if $(p, q) \notin E$, otherwise $W_{p,q} = 1$. In HIPRank, we normalize W and T, each row of W and T is normalized to one unless all elements in this row are zero, W' and T' denote the normalized W and T respectively. For the p-th row and q-th column element of W', $W'_{p,q} = 0$ if $(p, q) \notin E$, otherwise $W'_{p,q} = w(p, q)/\sum_{i \in O(p)} w(p, i)$, where $w(p, q)$ is the weight of edge (p,q) in W, the same rule to T'.

For the network depicted in Fig. 1, using HITS to rank nodes in it, the normalized *authority vector* of the nodes is [0.202,0.095,0.335,0.166,0.202] and the normalized *hub vector* of the nodes is [0.190,0.240,0.165,0.240,0.165]. HITS does not consider the initial importance of the nodes. That is, the initialized *authority* and *hub* are not propagated in HITS algorithm to rank nodes. The basic idea behind HITS is to measure the centrality (importance) of web pages in the web network using *authority* and *hub*, *authority* estimates the value of the content of a web page, and *hub* estimates the value of its links to other pages. The matrix form of HITS can be formulated as below in Eq. (1).

$$R_{a,k} = c_k R_{h,k-1} W; R_{h,k} = c'_k R_{a,k-1} T \tag{1}$$

Here, k represents the k-th iteration, c_k and c'_k are normalization factors. Based on Eq. (1), the *authority* and *hub* scores of HITS can be computed iteratively, and this process is proved convergent if we normalize the two score vectors of *authority* and *hub* after each iteration [5].

When considering a case that a user has some prior knowledge or preference on some nodes in Fig. 1 and prefers to retrieve nodes like node b, thus the initial importance, which represents the user's preference, may cast a crucial influence on the result of the ranking. However, in this case, typical HITS algorithm will not work because HITS does not take the initial importance of nodes into account when

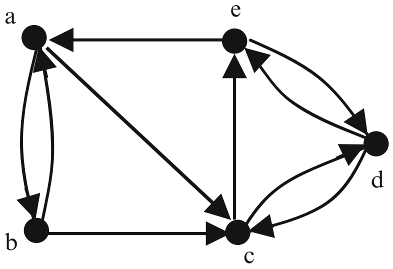

Fig. 1 A directed network consists of 5 nodes and every node has predefined importance in two aspects, including *authority* and *hub*

Table 2 Normalized HITS scores and HIPRank scores

$Z_a\&Z_h$	Normalized HIPRank Scores (%)
$Z_a = [0.2, \mathbf{0}, 0.2, 0.2, 0.2]$	$Authority$: [20.63, 8.75, 31.97, 18.02, 20.63]
$Z_h = [0.2, 0.2, 0.2, 0.2, 0.2]$	Hub : [17.92, 23.43, 17.61, 23.43, 17.61]
$Z_a = [0.2, \mathbf{0.2}, 0.2, 0.2, 0.2]$	$Authority$: [20.38, 9.23, 32.58, 17.41, 20.38]
$Z_h = [0.2, 0.2, 0.2, 0.2, 0.2]$	Hub : [18.58, 23.58, 17.13, 23.58, 17.13]
$Z_a = [0.2, \mathbf{0.4}, 0.2, 0.2, 0.2]$	$Authority$: [20.16, 9.68, 33.14, 16.86, 20.16]
$Z_h = [0.2, 0.2, 0.2, 0.2, 0.2]$	Hub : [19.20, 23.72, 16.68, 23.72, 16.68]
HITS scores	$Authority$: [20.2, 9.5, 33.5, 16.6, 20.2]
	Hub : [19.0, 24.0, 16.5, 24.0, 16.5]

The bold values indicate the authority score of node b

ranking nodes in the network. Additionally, HITS is originally designed to rank the web network. Simply applying it to rank nodes in general network may result in unexpected results, because the "random jumping" behavior of web network is not suitable for modeling some friendship based social networks.

To solve this problem, IPRank [6] is proposed to consider the initialized importance of nodes in the network. However, in other cases, predefining importance of all the nodes in only one dimension is not enough. For instance, in a paper citation network, some papers, such as surveys and reviews, cite a large number of other papers because they focus on reviewing the recent advancements in a domain. We may call this kind of papers *hub paper*. Meanwhile, other papers have a large number of citations because they focus on presenting specialized algorithms or pioneering approaches to some difficult problems in a domain. We may call this kind of papers *authority paper*. Simply using only one dimension to measure the predefined importance of these two kinds of papers in a citation network will cause a great loss of important information.

Motivated by the problems of HITS and IPRank, we propose a new influence propagation model called HIP to model bidirectional propagation of influences in the network, and a new ranking algorithm called HIPRank to rank nodes, based on the global structural contexts of the network accompanied with predefined importance of authority and hub. Table 2 shows the normalized HITS scores and HIPRank scores corresponding to different predefined importance Z_a and Z_h. The decay function is $f(k) = 0.8^k$.

3 Related Work

The related work of this paper can be categorized into two aspects. One is personalized PageRank. The basic idea is that while the global network topology inducing the adjacent matrix in PageRank is the same for all users, the preference vectors inducing users' preference on nodes are different for different users. However, a difficult problem of personalized PageRank is it needs huge computation. To address this problem, many solutions were proposed such as probabilistic random walk with external memory indexing [1], incremental computation [4] and top-K search with bounded accuracy [2]. Actually, in most cases, user preference is dependent on different domains and not a constant. Thus, it is very hard to capture user preference correctly due to its diversity and volatility.

This problem brings about the other related work of the paper, i.e. influence-propagation-based ranking methods. The basic idea is that for a search task in a given domain, a user has some prior knowledge of the influential nodes in the network. Thus, by propagating influence to other nodes based on the network topology, all the nodes in the work obtain their importance ranking. The methods in this aspect include IPRank with propagating influence based on PageRank [6], propagating relevance and irrelevance [9], propagating trust and distrust [3], etc. HIPRank also ranks nodes based on influence propagation. However, we are different from the previous work, we propagate *authority* and *hub* of nodes in the network by introducing the idea of HITS algorithm [5].

4 HIPRank

4.1 HIP Propagation Model

Let Z_a and Z_h be two vectors to represent predefined importance of *authority* and *hub* of nodes in G respectively, while Z_a represents the initialized *authority* values of nodes and Z_h represents the initialized *hub* values of nodes, and all the elements in Z_a and Z_h are non-negative.

In HITS algorithm, the *authority* of node c comes from the *hub* of its in-neighbors within 1-step hop and the *hub* of node c comes from the *authority* of its out-neighbors within 1-step hop. In HIP, the influence propagated bidirectionally. The *authority* of node c comes not only from the *hub* of its in-neighbors within 1-step hop, but also from the *hub* of its in-neighbors within k-step hop ($1 < k < K$, K is predefined as the maximum propagation step). The *hub* of node c comes not only from the *authority* of its out-neighbors within 1-step hop, but also from the *authority* of its out-neighbors within k-step hop ($1 < k < K$). For instance, assuming there is a path $p =< v_0, v_1, v_2, \ldots, v_k >$, the *hub* $Z_h(0)$ propagating from v_0 to v_k in forward direction, contributes to the *authority* of v_k. Equation (2) shows the received influence of v_k from v_0.

$$Z_a(0,k) = Z_h(0) \cdot f(k) \cdot \prod_{i=0}^{k-1} W'_{i,i+1} \tag{2}$$

The authority $Z_a(k)$ propagating from v_k to v_0 in backward direction, contributes to the *hub* of v_0. Equation (3) shows the received influence of v_0 from v_k.

$$Z_h(k,0) = Z_a(k) \cdot f(k) \cdot \prod_{i=0}^{k-1} T'_{i,i+1} \tag{3}$$

We introduce a discrete *decay function* $f(k) = c^k$ to capture the retained influence on the k-th step hop. Here $k \in \{1, 2, 3, \ldots, K\}$, K is predefined as the maximum propagation step, and $0 < c < 1$. Generally, $f(k) < 1$, and the lager k results in smaller $f(k)$. In order to decide the maximum propagation step K, a threshold h that satisfies the following condition needs to be specified.

$$f(K) \geq h \; and \; f(K+1) < h \tag{4}$$

Proposition 1. *Without decay function, that means when $f(k)$ is a constant, influence also decays in HIP model.*

Proof. According to Eqs. (2) and (3), when $f(k)$ is a constant, here denoted by C, the *authority* influence propagating from v_0 to v_k can be calculated by $Z_a(0,k) = Z_h(0) \cdot C \cdot \prod_{i=0}^{k-1} W'_{i,i+1}$, since each row in W' is normalized to one, so each element in W' is less than 1 when $k \to \infty$ $\prod_{i=0}^{k-1} W'_{i,i+1} \to 0$, thus the propagated influence decays via the propagation path. Similarly, the same with *hub* influence. Proposition 1 holds.

The reason why we introduce decay function $f(k)$ in HIP model is that in HIP the decay of influence is determined by the topological structure of the network, which is uncontrolled by users. With a specified decay function, a user can control the propagation of influence over nodes.

4.2 HIPRanking Nodes

Based on the HIP propagation model, the HIPRank scores of nodes in the network can be defined as follows.

Definition 1. The HIPRank scores of a node in the network consist of a *authority* score and a *hub* score, and both scores are measured by the initialized importance of this node and the influence propagated to this node from other nodes.

The basic idea behind HIPRank is that, the more influence of *authority* or *hub* a node receives from other nodes, the more authoritative or directive the node is in the network. Different from IPRank [6], the influence propagation in HIPRank considers initial importance of a node in two dimensions: *authority* and *hub*. The initial *authority* and *hub* of nodes are represented by Z_a and Z_h respectively. Both Z_a and Z_h are propagated in HIP to impact *authority* and *hub* scores of nodes as shown in Table 2.

The *authority* and *hub* of nodes in HIPRank can be computed in the same manner. The only difference between *authority* and *hub* lies in that, the *authority* of a node propagates in the backward direction, and the *hub* of a node propagates in the forward direction. For this reason, we use the computation of *authority* score of a node in HIPRank as an example to show how the HIPRank algorithm ranks nodes in the network.

With the decay during the propagation, the propagated influence of *hub* and *authority* can be ignored after K steps. Therefore, we only need to collect the propagated influences that reach a node within K steps with random walk [7, 10].

Considering the node a in the graph G in Fig. 1 and supposing $K = 1$, that is, the influence of *authority* and *hub* propagates within 1 step. To compute the *authority* score of node a, we reverse all edges and traverse b and e starting from the node a. And two random walk paths (b, a) and (e, a) in G that reach node a within 1 step hop are collected. Thus, the 1 step *authority* propagation from node b and e to a as $Z_{a,1}(a)$ can be denoted in Eq. (5), where Z_h(b) and Z_h(e) represent the hub scores of node b and e, respectively.

$$Z_{a,1}(a) = f(1) \cdot (Z_h(b)W'_{b,a} + Z_h(e)W'_{e,a}) \tag{5}$$

The *authority* score of node a in Fig. 1 can be calculated in Eq. (6), where Z_a(a) represents the initial *authority* score of node a.

$$R_a(a) = Z_{a,1}(a) + Z_a(a) \tag{6}$$

Note that, different random paths generated by HIPRank may have common sub-paths, which could be reused to save computational cost. For example, *hub* propagation along paths $< c, d, c, e >$ and $< c, d, c >$ are generated by HIPRank queries for node e and c, and these two paths share the common sub-path $< c, d, c >$, which can be reused in practical computation.

For all the nodes in a network, we develop an algorithm, called *HIPRank-All* to compute their HIPRank scores in matrix form as shown in Algorithm 1.

In HIPRank, the initial *authority* and *hub* of all nodes are stored in two vectors Z_a and Z_h respectively. Taking the computation of *authority* score as an example, in the first step, all the nodes propagate *hub* to their out-neighbors with decay factor $f(1)$. Considering the *hub* received by a node which contributes to its *authority*, let $I_1(v)$ be the in-neighbor set of node v within 1 step hop. The *hub* received by v is $Z_{a,1}(v) = f(1) \cdot \sum_{n=1}^{I_1(v)} Z_h(n)W'_{n,v}$. Further, considering all the nodes in the network, we obtain $Z_{a,1} = f(1) \cdot Z_h W'$ in matrix form. In the second step, all the nodes that are 2-step in-neighbors to node v, represented by $I_2(v)$, will also propagate *hub* to v and contribute to the *authority* of v. Thus, the *authority* of v obtained from the second step is $Z_{a,2}(v) = f(2) \cdot \sum_{n=1}^{I_2(v)} Z_h(n)W'^2_{n,v}$. Considering all the nodes in the network, we obtain $Z_{a,2}(v) = f(2) \cdot Z_h W'^2_{n,v}$ in matrix form. By analogy, the obtained *authority* vector of all nodes in the network on the k-th step can be computed by Eq. (7).

$$Z_{a,k} = f(k) \cdot Z_h W'^k \tag{7}$$

```
Algorithm 1 HIPRank-All($G, Z_a, Z_h, h, W, T$)
Input: Graph $G(V, E)$, initial authority vector $Z_a$, initial hub vector $Z_h$, threshold $h$ for deciding
    the maximum propagation step, $W$ is the adjacency matrix of $G$, and $T$ is the transpose of $W$.
Output: HIPRank scores $R_a$ and $R_h$
initialize $R_a = Z_a$; $R_h = Z_h$;
obtain $W^{'}$ and $T^{'}$ by normalizing $W$ and $T$;
for every node $v \in V$ do
    obtain $K$ according to Equation (4);
    AuthorityRecursion($v, Z_h(v), 0, K$);
    HubRecursion($v, Z_a(v), 0, K$);
end for
return $R_a$ and $R_h$;
procedure AUTHORITYRECURSION($v, x, y, K$)
    $y = y + 1$;
    for every node $u$ in out-neighbor set of node $v$ do
        $R(u) = R(u) + x \cdot W^{'}_{v,u} \cdot f(y)$
        if $y < K$ then
            AuthorityRecursion($u, x \cdot W^{'}_{v,u}, y, K$);
        end if
    end for
end procedure
procedure HUBRECURSION($v, x, y, K$)
    $y = y + 1$;
    for every node $u$ in out-neighbor set of node $v$ do
        $R(u) = R(u) + x \cdot T^{'}_{v,u} \cdot f(y)$
        if $y < K$ then
            HubRecursion($u, x \cdot T^{'}_{v,u}, y, K$);
        end if
    end for
end procedure
```

As a result from Definition 1, in HIPRank, the *authority* ranking vector obtained within k steps can be described in Equation (8).

$$R_{a,k} = \sum_{i=1}^{k} Z_{a,i} + Z_a = \sum_{i=1}^{k} f(i) \cdot Z_h W^{'i} + Z_a \tag{8}$$

In the same manner, we can obtain the *hub* ranking vector within k steps, as described in Eq. (9).

$$R_{h,k} = \sum_{i=1}^{k} Z_{h,i} + Z_h = \sum_{i=1}^{k} f(i) \cdot Z_a T^{'i} + Z_h \tag{9}$$

Equations (8) and (9) describe the main computation of *HIPRank-All* algorithm. The time complexity of *HIPRank-All* algorithm is $O(KP)$, P is the number of edges traversed in the propagation process. One useful proposition about HIPRank computing is given below.

Proposition 2. *When $f(k) = c^k (0 < c < 1)$ and $k \to \infty$, HIPRank is convergent.*

Proof. According to Eq. (8), since each row in W'^k is normalized to one, when $k \to \infty$, $W'^k \to 0$. The decay function is $f(k) = c^k (0 < c < 1)$, therefore: $R_{a,k} = Z_h\big(cW' + c^2W'^2 + c^3W'^3 + \ldots + c^kW'^k\big) + Z_a$

$$k \to \infty \quad R_a = Z_h(\mathbf{E} - cW')^{-1} + Z_a \tag{10}$$

Similarly:

$$R_h = Z_a(E - cT')^{-1} + Z_h \tag{11}$$

Here, **E** is the identity matrix and **0** is the zero matrix. Equations (10) and (11) show that when $f(k) = c^k (0 < c < 1)$ and $k \to \infty$, R_a and R_h are convergent. Proposition 2 holds.

In HITS the only factor that influences the final *authority* and *hub* scores is the topological structure of the network, while in HIPRank, from Eqs. (10) and (11), we can see that both the topological structure of the network and the initialized importance of *authority* and *hub* can influence the final ranking scores.

5 Experiments

5.1 Appropriate Propagation Step K

For obtaining acceptable scores for all the nodes, an appropriate propagation step K should be specified. We conduct a simulation on a PC with a 3.4 GHz CPU and an 8GB RAM to exam how to set an appropriate K. We use three random networks called G_1(1 million nodes and 3 million edges), G_2(1 million nodes and 5 million edges) and G_3(1 million nodes and 10 million nodes). Similar to Pei Li et al [6], we also use the precision defined by average $R_k(a)/R(a)$ to observe the convergence rate of HIPRank on those three graphs. For G_1 we set the $|E|/|V| = 3$, Fig. 2a shows that the error of precision is below 0.01 after 9 iterations; for

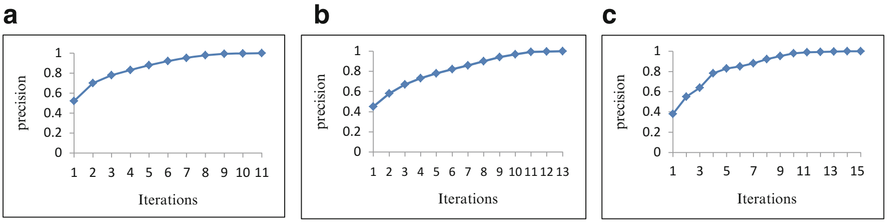

Fig. 2 Convergence rate of three networks (left to right G_1, G_2 and G_3)

G_2, $|E|/|V| = 5$, results in Fig. 2b shows after 11 iterations the error of precision is below 0.01; for G_3, $|E|/|V| = 10$, Fig. 2c shows that the error of precision is below 0.01 after 12 iterations. So, with the proportion of the number of edges and nodes increasing, HIPRank needs more iteration to obtain high convergence rate. In Fig. 2 when $K = 10$, the precision of G_1 is 0.996, G_2 is 0.97 and G_3 is 0.95, so we recommend when $|E|/|V| < 10$, the max propagation step K is set to 10 and when $|E|/|V| \geq 10$, the K should be bigger than 10.

5.2 *Time Cost of HIPRank*

The time complexity of *HIPRank-All* algorithm is $O(KP)$, where K is the maximum propagation steps and P is the number of edges traversed in the propagation process. Figure 3a shows the time cost when K increases. We can see that when K is smaller than 9, which means the influence of a node propagates less than 9 steps, the time cost of HIPRank keeps very small because the number of traversed edges in the propagation process would be not very large at this duration. However, when K is larger than 10, there is a linear increase in the time cost as we explain that all the edges in the network are traversed in the propagation process. Figure 3b shows the time cost when M increases. We set K as 10 to obtain high convergence rate. As M increases, P increases at the same time, and the time cost of HIPRank increases linearly when M increase from 0.1 to 0.7 million.

In practice, P is usually much larger than K. Thus, the computation complexity is almost decided by P, which is not larger than the number of edges (M) in the work. In this case, traversed edges in the proposed HIP rank model are much smaller than that in models proposed by [2] and [9] to find the top-K relevant nodes.

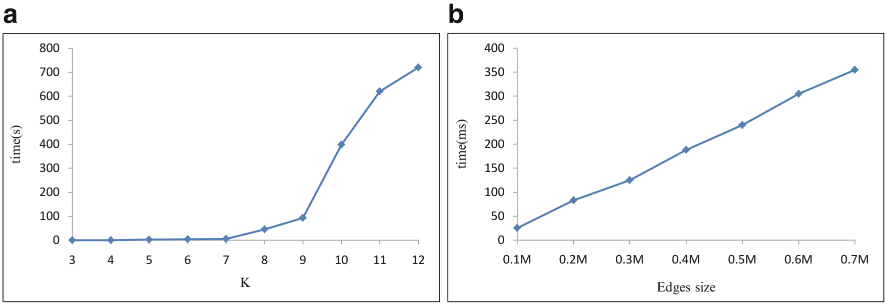

Fig. 3 (**a**) Time cost when K increases ($N = 1.5$ million, $M = 3.5$ million). (**b**) Time cost when edges size increases ($K = 10$, $N = 0.1$ million)

5.3 Results on DBLP Paper Citation Network

We build a large paper citation network using the citation information of the entire DBLP conference papers. This network consists of 1,511,035 papers (nodes) and 2,084,019 citations (edges).

Three expected outcomes of the experiment are: (i) if a user has already known that some papers are important in some fields, for example, we know that the paper "C4.5: Programs for Machine Learning" is an authoritative paper in KDD area, a user can find other authoritative and directory papers (*authority paper* and *hub paper*) in KDD area using the HIPRank model by setting high *authority* scores to this paper; (ii) Papers with high *authority* scores have larger possibilities to propose novel algorithms or pioneering methods to solve difficult problems in a domain. We can call these papers *authority paper*; (iii) Papers with high *hub* scores have larger possibilities to be surveys, overviews, or reviews, which called *hub paper*. The decay function is $f(k) = 0.8^k$ and the maximum propagation step is set to 10. The initial *authority* and *hub* values of the predefined nodes are set to $N/2$, the initial values of the rest nodes are set to $1/N$, and then we normalize all the initial values to the range from 0 to 1 for all the nodes, where N is the number of nodes in the network.

First we use HITS on the paper citation network, and the corresponding top-10 *authority paper* and top-10 *hub paper* are shown in Table 3a. For the top-10 *authority paper*, papers rank high only because they have high citations; for the top-10 *hub paper*, papers rank high mainly because they cite *authority paper*.

Second, we bias the ranking to a special area by predefining importance (both *authority* and *hub*) for papers. In Table 3b, papers published in SE (Software Engineering) area conferences (here we use ICSE, FSE, ESEM, and SIGSOFT) are given higher predefined *authority* and *hub* scores. Then we obtain top-10 *authority paper* and top-10 *hub paper* in Software Engineering area.

Third, we bias HIPRank to KDD(Knowledge Discovery and Data Mining) area (here we use conferences: SIGKDD, PAKDD, PKDD, and ICDM) and show results in Table 3c. Those *authority paper* and *hub paper* of special areas listed in Table 3b,c show that our HIPRank with predefined *authority* and *hub* scores produces reasonable results.

From Table 3b,c, we can see that many papers which are not initialized with relatively high *authority* and *hub* are retrieved from DBLP dataset, such as those papers from ACM TOSEM for SE area and SGIMOD for KDD area. These outcomes illustrate that HIPRank is enlightening in discovering papers by user preference.

Table 3 (**a**) Results of HITS. (**b**) HIPRank on SE area. (**c**) HIPRank on KDD area

(a)			
Top10 *Authority papers*	Conf	Top10 *Hub papers*	Conf
C4.5: Programs for Machine Learning	BOOK	Data Mining: An Overview from a Database Perspective	IEEE Trans
Fast Algorithms for Mining Association Rules in Large Databases	VLDB	Scalable Algorithms for Mining Large Databases	SIGKDD
Mining Association Rules between Sets of Items in Large Databases	SIGMOD	Mining Query Logs: Turning Search Usage Data into Knowledge	FTIR
Introduction to Algorithms	BOOK	Scalable frequent-pattern mining methods: an overview	SIGKDD
Introduction to Modern Information Retrieval	BOOK	ACIRD: Intelligent Internet Document Organization and Retrieval	IEEE Trans
Modern Information Retrieval	BOOK	ART: A Hybrid Classification Model	Machine Learning
Induction of Decision Trees	BOOK	Ratio Rules: A New Paradigm for Fast, Quantifiable Data Mining	VLDB
Compilers: Princiles, Techniques, and Tools	BOOK	Using Information Retrieval techniques for supporting data mining	Data Knowledge Engineering
The Anatomy of a Large-Scale Hypertextual Web Search Engine	WWW	Association Rule Mining, Models and Algorithms	BOOK
Mining Frequent Patterns without Candidate Generation	SIGMOD	From intra-transaction to generalized inter-transaction: Landscaping multidimensional contexts in association rule mining	Information Sciences
(b)			
The Model Checker SPIN	IEEE Trans	Research Directions in Requirements Engineering	FOSE
Communicating Sequential Processes	Communications of the ACM	A brief survey of program slicing	SIGSOFT
Statecharts: A Visual Formalism for Complex Systems	Science of computer programming	Architecture Reconstruction	SE
Experiments of the Effectiveness of Dataflow and Controlflow-Based Test Adequacy Criteria	ICSE	Requirements interaction management	CSUR
Compilers: Princiles, Techniques, and Tools	BOOK	A schema for interprocedural modification side-effect analysis with pointer aliasing	TOPLAS

(continued)

Table 3 (continued)

Top10 *Authority papers*	Conf	Top10 *Hub papers*	Conf
A Formal Basis for Architectural Connection	TOSEM	Software Unit Test Coverage and Adequacy	CSUR
Software Processes Are Software Too	ICSE	Context-aware statistical debugging: from bug predictors to faulty control flow paths	ASE
Automatic Verification of Finite-State Concurrent Systems Using Temporal Logic Specifications	TOSEM	The IBM-McGill project on software process	CASCON
Bandera: extracting finite-state models from Java source code	SE	Profile-guided program simplification for effective testing and analysis	SIGSOFT
(c)			
Fast Algorithms for Mining Association Rules in Large Databases	VLDB	Scalable frequent-pattern mining methods: an overview	SIGKDD
C4.5: Programs for Machine Learning	BOOK	Scalable Algorithms for Mining Large Databases	SIGKDD
Mining Association Rules between Sets of Items in Large Databases	SIGMOD	Data Mining: An Overview from a Database Perspective	IEEE Trans
Mining Frequent Patterns without Candidate Generation	SIGMOD	From intra-transaction to generalized inter-transaction: Landscaping multidimensional contexts in association rule mining	Information Sciences
Mining Sequential Patterns	BOOK	Association Rule Mining, Models and Algorithms	BOOK
Induction of Decision Trees	BOOK	Off to new shores: conceptual knowledge discovery and processing	Human Computer Studies
An Efficient Algorithm for Mining Association Rules in Large Databases	VLDB	A template model for multidimensional inter-transactional association rules	VLDB
An Effective Hash Based Algorithm for Mining Association Rules	SIGMOD	Efficient dynamic mining of constrained frequent sets	TODS
Dynamic Itemset Counting and Implication Rules for Market Basket Data	SIGMOD	Mining Frequent Patterns without Candidate Generation: A Frequent-Pattern Tree Approach	Data Mining Knowledge Discovery
Mining Quantitative Association Rules in Large Relational Tables	SIGMOD	ART: A Hybrid Classification Model	Machine Learning

Conclusion
This paper proposes a new ranking model, called HIPRank, to rank individuals based on their influence propagation of *authority* and *hub* in the network. The basic idea of HIPRank is to make use of user preference and prior knowledge of nodes to initialize the *authority* and *hub* of nodes. Then, the initialized *authority* and *hub* of each node are propagated to other nodes through the topology of the network. Finally, the importance of each node in the network is measured by summing their initialized *authority* and *hub* with the propagated *authority* and *hub* from other nodes within the predefined K-step hops. Also, users can control the propagation by defining decay function and the maximum propagation step K. Experiments on synthetic data and the real DBLP citation dataset demonstrate the effectiveness of the proposed approach in retrieving user-intended authoritative and directory individuals from the network.

Acknowledgements This research was supported in part by National Natural Science Foundation of China under Grant Nos. 61073044, 71101138, 61003028 and 61379046; National Science and Technology Major Project under Grant Nos. 2012ZX01039-004; Beijing Natural Science Fund under Grant No.4122087; State Key Laboratory of Software Engineering of Wuhan University.

References

1. Csalogny, K., et al.: Towards scaling fully personalized pagerank: algorithms, lower bounds, and experiments. Internet Math. **2**(3):333–358 (2005)
2. Fujiwara, Y., et al.: Efficient personalized pagerank with accuracy assurance. In: Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 15–23 (2012)
3. Guha, R.V., et al.: Propagation of trust and distrust. In: Proceedings of the 13th International Conference on World Wide Web, pp. 403–412 (2004)
4. Jeh, G., Widom, J.: Scaling personalized web search. In: Proceedings of the 12th International Conference on World Wide Web, pp. 271–279 (2003)
5. Kleinberg, J.M.: Authoritative sources in a hyperlinked environment. J. ACM. **46**(5):604–632 (1999)
6. Li, P., et al.: Ranking individuals and groups by influence propagation. In: Proceedings of the 15th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining, pp. 407–419 (2011)
7. Motwani, R., Raghavan, P.: Randomized Algorithms. Cambridge University Press, Cambridge (1995)
8. Page, L., Brin, S., Motwani, R., Winograd, T.: The pagerank citation ranking: bringing or-der to the web. Technical Report 1999-66, Stanford InfoLab (1999)
9. Sarkar, P., Moore, A.W.: Fast dynamic reranking in large graphs. In: Proceedings of the 18th International Conference on World Wide Web, pp. 31–40 (2009)
10. Valente, T.: Network Models of the Diffusion of Innovations. Hampton Press, Cresskill (1995)